

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Kiyosi ITO et al.

Group Art Unit:

Serial No.:

Examiner:

Filed: March 13, 2001

For: COMPILER PARALLELIZING SCHEDULE METHOD

1046 U.S. PRO
09/804031
03/13/01



**SUBMISSION OF CERTIFIED COPY OF PRIOR
FOREIGN APPLICATION IN ACCORDANCE WITH
THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application(s):

Japanese Patent Application No. 2000-309759
Filed: October 10, 2000

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date, as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,
STAAS & HALSEY LLP

Date: March 13, 2001

By: 

H. J. Staas
Registration No. 22,010

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

J1046 U.S. PTO
09/804031
03/13/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2000年10月10日

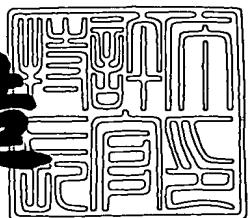
出願番号
Application Number: 特願2000-309759

出願人
Applicant(s): 富士通株式会社

2001年 2月 9日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3005763

【書類名】 特許願

【整理番号】 0000912

【提出日】 平成12年10月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16
G06F 9/45

【発明の名称】 コンパイラ並列化スケジュール方法

【請求項の数】 9

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 伊藤 清志

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 富居 義仁

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 岩間 芳樹

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 植草 直幸

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 佐藤 恒央

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100104190

【弁理士】

【氏名又は名称】 酒井 昭徳

【手数料の表示】

【予納台帳番号】 041759

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9906241

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンパイラ並列化スケジュール方法

【特許請求の範囲】

【請求項1】 命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【請求項2】 命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

同じプライオリティ値の命令間に発行制限による遅延があるか否かを調べる工程と、

発行制限による遅延がある場合に、前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

前記新プライオリティ値に基づいて命令の発行順序を決定し、前記各命令のスロット配置をおこなう工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【請求項3】 発行制限による遅延がある命令群を最適化対象グループとし、当該最適化対象グループに含まれる複数の命令に共通の先行命令をネック命令とし、当該ネック命令と前記最適化対象グループとの間で前記逆プライオリティ値を求めるなどを特徴とする請求項2に記載のコンパイラ並列化スケジュール方法。

【請求項4】 前記重み付けは、前記最適化対象グループから前記ネック命

令までの命令に付与される第1の重み値と、前記ネック命令よりも先の先行命令に付与される第2の重み値とからなることを特徴とする請求項3に記載のコンパイラ並列化スケジュール方法。

【請求項5】 前記最適化対象グループに含まれる複数の命令に対して、前記逆プライオリティ値の小さい順、前記逆プライオリティ値が同じ場合には前記先行命令数の少ない順、前記逆プライオリティ値と前記先行命令数が同じ場合には行番号の早い順、または前記逆プライオリティ値と前記先行命令数と前記行番号が同じ場合には生成順番の早い順に、優先順位を設定し、その優先順位にしたがって、前記第1の重み値を決定することを特徴とする請求項4に記載のコンパイラ並列化スケジュール方法。

【請求項6】 前記優先順位にしたがって、第1番目の命令の第1の重み値を、前記最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数から1を引いた値とし、第2番目以降の命令の第1の重み値を前記命令数から1を引いた値から順次1ずつ小さくした値とすることを特徴とする請求項5に記載のコンパイラ並列化スケジュール方法。

【請求項7】 前記最適化対象グループ内の各命令に対する先行命令の第1の重み値を、当該先行命令につづく後続命令の第1の重み値をそのまま継承した値とし、前記後続命令が複数存在する場合には最も大きい第1の重み値を継承した値とすることを特徴とする請求項5または6に記載のコンパイラ並列化スケジュール方法。

【請求項8】 前記ネック命令の先行命令の第2の重み値を、前記ネック命令に対応する最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数に等しい値とすることを特徴とする請求項4に記載のコンパイラ並列化スケジュール方法。

【請求項9】 前記ネック命令の先行命令の第2の重み値を、前記ネック命令に対応する最適化対象グループとは別の最適化対象グループに起因して新たな第2の重み値が発生した場合には、当該第2の重み値を加算した値とすることを特徴とする請求項4に記載のコンパイラ並列化スケジュール方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンパイラ並列化スケジュール方法に関し、特にV L I W (Very Long Instruction Word) アーキテクチャ用コンパイラの並列化スケジュール方法に関する。

【0002】

従来より、並列に動作し得る複数の演算ユニットを持つ計算機（コンピュータ）用の目的コードを生成するコンパイラにおいて、目的コードの効率を高めるため、複数の演算ユニットにおいて異なる命令を並列に処理するようなスケジューリングがおこなわれている。

【0003】

【従来の技術】

従来、コンパイラ並列化スケジュール方法として、特開平10-207854号公開公報に記載されたものがある。このスケジュール方法は、命令間の依存関係と並列動作抑制関係を依存・並列動作抑制グラフで表し、それに基づいてパスレイテンシおよび並列動作抑制数を計算し、その計算結果が追加された依存・並列動作抑制グラフに基づいて目的コードの並列なスケジューリングをおこなうものである。

【0004】

【発明が解決しようとする課題】

しかしながら、上述した従来のスケジュール方法では、全命令のノードに対して並列動作抑制関係と並列動作抑制数を調べるため、その処理に時間がかかるてしまい、目的コードの並列化処理に要する時間が長くなるという欠点がある。また、上述した従来のスケジュール方法では、命令に付与する重み値が並列動作抑制数の1種類であるため、並列度が十分に高いとはいせず、さらなる並列度の向上を図る余地がある。

【0005】

本発明は、上記問題点に鑑みてなされたものであって、並列に動作し得る複数の演算ユニットを持つ計算機用の目的コードを生成するコンパイラにおいて、目

的コードの並列度を向上させるとともに、より高速に目的コードの並列化処理をおこなうことができるコンパイラ並列化スケジュール方法を提供することを目的とする。

【0006】

【課題を解決するための手段】

上記目的を達成するため、本発明にかかるコンパイラ並列化スケジュール方法は、従来のプライオリティ値ごとに命令を分類し、プライオリティ値が同じ命令間で発行制限があるか否かを調べ、さらに発行制限がある命令群について発行制限による遅延があるか否かを調べ、発行制限による遅延がある命令群よりなる最適化対象グループから、その最適化対象グループ内の命令に共通の先行命令であるネック命令までの命令に対して逆プライオリティ計算をおこない、その逆プライオリティ値に基づいて、最適化対象グループからネック命令までの命令に第1の重み値（アドバンテージ）を付与し、一方、ネック命令よりも先の先行命令に第2の重み値（ウェイト）の重み値を付与し、それら重み値を考慮して再度プライオリティ計算をおこない、その新プライオリティに基づいて各命令に対してスロット配置をおこなうことを特徴とする。

【0007】

この発明によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群について発行制限による遅延があるか否かを調べるために、全命令のノードに対して並列動作抑制関係を調べるのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【0008】

また、逆プライオリティ計算および第1の重み値（アドバンテージ）を付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、全命令のノードに対して並列動作抑制数を計算するのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【0009】

さらに、最適化対象グループからネック命令までの命令に第1の重み値（アド

バンテージ) を付与し、ネック命令よりも先の先行命令に第2の重み値(ウェイト)を付与するため、命令に付与する重み値が並列動作抑制数だけである場合に比べて、目的コードの並列度を向上させることができる。

【0010】

【発明の実施の形態】

以下に、本発明の実施の形態について図1～図10を参照しつつ詳細に説明する。

【0011】

(処理の概要の説明)

まず、本発明にかかるコンパイラ並列化スケジュール方法の概要について、単純化したモデルを例にして説明する。図1は、本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャートである。また、図2～図8は、本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図または図表である。

【0012】

まず、命令間の依存関係を表す有向グラフ(以下、DAGとする。DAG: directed acyclic graph)を生成し(ステップS11)、プライオリティ計算をおこなう(ステップS12)。プライオリティ計算をおこなうにあたっては、まず、ステップS11で生成したDAGに基づいて従来のプライオリティ計算をおこなう(ステップS15)。

【0013】

図2に、説明のため単純化したサンプルソースのモデルと、そのソースについてDAG生成およびプライオリティ計算をおこなった結果を示す。図2において、中央の矢印を挟んで左側がモデルとなるサンプルソースであり、右側がそのモデルのDAGおよびプライオリティ値を示している。

【0014】

図2に示すモデルにおいて、命令1(図に①で示す)は命令2(図に②で示す)と依存関係にあり、また命令3(図に③で示す)は命令4(図に④で示す)と依存関係にある。命令5(図に⑤で示す)と他の命令との間には依存関係がない

。DAGにおいて、命令1と命令2を結ぶ線、および命令3と命令4を結ぶ線はそれぞれ前述した依存関係を表している（他の図のDAGにおいても同じ）。さらに、命令2、命令4および命令5はストア命令であり、たとえば2個のI0スロット、I1スロットのうちI0スロットでしか発行することができないという制限があると仮定する。

【0015】

従来のプライオリティ計算の結果、命令2、命令4および命令5のプライオリティ値は1、命令1および命令3のプライオリティ値は2である。これは、従来のプライオリティ計算では、後続の命令を従えていない命令のプライオリティ値を1とし、後続の命令から先行命令の方向に順にたとえば1ずつ加算していくからである。ここで、加算する値が1であるのは、ある命令のプライオリティ値は、その命令の後続命令のプライオリティ値に発行レイテンシとペナルティを加算して求めるが、ここでは単純化のため発行レイテンシを1、ペナルティを0（ゼロ）と仮定しているからである。なお、DAGにおいて、各命令のノードを表す①～⑤の右下の数字は各命令のプライオリティ値を表す。

【0016】

図1に戻り、従来のプライオリティ計算（ステップS15）につづいて、発行制限の最適化処理を実行するか否かの判断をおこなう（ステップS16）。具体的には、図3に示すように、従来のプライオリティ計算で得られたプライオリティ値ごとに命令を分類したプライオリティ表を作成する。そして、同じプライオリティ値の命令を、スロットの配置制限を考慮して配置するのに要するVLIW命令数（これを、実VLIW命令数とする）を求める。

【0017】

また、同じプライオリティ値の命令数と同じ数の命令を、スロットの配置制限がない場合に発行するのに必要なVLIW命令数（これを、最小VLIW命令数とする）を求める。ここで、最小VLIW命令数はつぎの（1）式より求められる。

【0018】

$$[\text{最小VLIW命令数}] = [\text{同じプライオリティ値の命令数}] / [\text{スロット数}]$$

+ [余り] . . . (1)

【0019】

実VLIW命令数と最小VLIW命令数とを比較し、同じプライオリティ値の命令間に発行制限があるか否かを調べる。実VLIW命令数と最小VLIW命令数との比較の結果、つぎの(2)式を満たす場合に発行制限の最適化処理が必要であると判断する。(2)式を満たさない場合に発行制限の最適化処理は不要であると判断する。

【0020】

[実VLIW命令数] > [最小VLIW命令数] . . . (2)

【0021】

図2に示すモデルの場合、図4に示すように、プライオリティ値が2の命令群では、実VLIW命令数および最小VLIW命令数はともに1であり、上記(2)式を満たさないので、発行制限の最適化処理は不要である。しかし、プライオリティ値が1の命令群では、実VLIW命令数は3であり、それに対して最小VLIW命令数は2であるため、上記(2)式が成り立つ。したがって、このモデルの場合、発行制限の最適化処理が必要となる。

【0022】

図1に戻り、発行制限の最適化処理をおこなう(ステップS17)。具体的には、まず、各命令の最短命令終了時刻を意味する逆プライオリティ値を求める。逆プライオリティ値は、先行命令がない命令のプライオリティ値を逆プライオリティ1とし、その命令から後続命令の方向に順にたとえば1ずつ加算することにより求められる。この計算を逆プライオリティ計算と呼ぶ。ここで、加算する値が1であるのは、ある命令の逆プライオリティ値は、つぎの(3)式より求められるが、ここでは単純化のため現在のDAGのレイテンシを1とし、先行命令と現在のDAGのペナルティを0(ゼロ)と仮定しているからである。

【0023】

[逆プライオリティ値] = [先行する命令群の中で最大の逆プライオリティ値]
+ [現在のDAGのレイテンシ] + [先行命令と現在のDAGのペナルティ]
. . . (3)

【0024】

図2に示すモデルに対する逆プライオリティ計算の結果を図5に示す。図5に示すDAGにおいて、各命令のノードを表す①～⑤の右下の数字は各命令の従来のプライオリティ値であり、左下の数字は各命令の逆プライオリティ値である。図5に示すように、命令1、命令3および命令5の逆プライオリティ値は1であり、命令2および命令4の逆プライオリティ値は2である。

【0025】

つづいて、発行制限があるプライオリティ値が1の命令群、すなわち命令2、命令4および命令5について重み値を付ける。その際、逆プライオリティ値が最小の命令に対して、実VLIW命令数から1を引いた値を重み値として付与する。そして、逆プライオリティ値が大きくなるにつれて順に重み値を1ずつ小さくする。なお、逆プライオリティ値が同じ命令どうしでは、生成が早いものの方が重み値が大きくなるようにする。

【0026】

図2に示すモデルについて、実VLIW命令数が3であるため、図6にまとめて示すように、逆プライオリティ値が一番小さい命令5の重み値は2である。また、逆プライオリティ値が2である命令2と命令4のうち生成順が早い命令2の重み値は1である。したがって、命令4の重み値は0（ゼロ）である。

【0027】

つづいて、図1に戻り、発行制限がある命令群に含まれる各命令に付与された重み値を考慮して、再度プライオリティ計算をおこない、新プライオリティ値を取得する（ステップS18）。ここで、新プライオリティ値はつぎの（4）式により得られる。

【0028】

[新プライオリティ値] = [後続命令のプライオリティ値] + [発行レイテンシ] + [ペナルティ] + [重み値] . . . (4)

【0029】

なお、上述したように、このモデルでは、単純化するため、発行レイテンシを1とし、ペナルティを0（ゼロ）としている。図2に示すモデルについて、図7

に示すDAGの各命令のノードを表す①～⑤の右下の数字が各命令の新プライオリティ値である。すなわち、命令1および命令5の新プライオリティ値は3である。命令2および命令3の新プライオリティ値は2である。命令4の新プライオリティ値は1である。

【0030】

図1に戻り、各命令の新プライオリティ値に基づいて、レディリストを生成し（ステップS13）、スロット配置をおこない（ステップS14）、スケジューリングを終了する。図2に示すモデルでは、図8に示すように、命令5、命令1、命令2、命令3、命令4の順にスロット配置がおこなわれる。したがって、コンパイラの内部テーブル（擬似機械テーブル）において、あるサイクルにおいてI0スロットおよびI1スロットからそれぞれ命令5および命令1が発行され、つぎのサイクルでI0スロットおよびI1スロットからそれぞれ命令2および命令3が発行され、さらにつぎのサイクルでI0スロットから命令4が発行される配置となる。

【0031】

ところで、ステップS16で発行制限の最適化処理を実行しないと判断した場合には、ステップS15で計算した旧プライオリティ値をそれ以降の処理のプライオリティ値とし（ステップS19）、それを用いてレディリストの生成（ステップS13）およびスロット配置（ステップS14）をおこない、スケジューリングを終了する。

【0032】

（処理の詳細な説明）

つぎに、本発明にかかるコンパイラ並列化スケジュール方法について、より具体的なモデルを例にして詳細に説明する。図9は、本発明にかかるコンパイラ並列化スケジュール方法を詳細に示すフローチャートである。また、図10～図19は、本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図または図表である。

【0033】

まず、処理の概要において説明したように、DAGを生成し、従来のプライオ

リティ計算をおこなう（ステップS91）。図10に、モデルとなるDAGおよび各命令のプライオリティ値の例を示す。このモデルでは、命令1（図に①で示す）は命令2（図に②で示す）、命令4（図に④で示す）および命令6（図に⑥で示す）と依存関係にある。また、命令2は命令3（図に③で示す）と依存関係にある。命令4は命令5（図に⑤で示す）と依存関係にある。さらに、命令3、命令5および命令6はストア命令であり、たとえば2個のI/Oスロット、I1スロットのうちI/Oスロットでしか発行することができないという制限があるとする。また、単純化するため、発行レイテンシおよびペナルティをそれぞれ1および0（ゼロ）とする。

【0034】

従来のプライオリティ計算の結果、命令1のプライオリティ値は3、命令2および命令4のプライオリティ値は2、命令3、命令5および命令6のプライオリティ値は1である。従来のプライオリティ計算については処理の概要において説明したので、省略する。なお、図10に示すDAGにおいて、各命令のノードを表す①～⑥の右下の数字は各命令のプライオリティ値を表す。つづいて、図9に戻り、従来のプライオリティ値ごとに命令を分類してプライオリティ表を作成する（ステップS92）。作成したプライオリティ表を図11に示す。

【0035】

図9に戻り、プライオリティ値が1、2および3の各グループについて、そのグループに属する命令間で発行制限があるか否かを調べる（ステップS93）。図10に示すモデルでは、図12に示すように、プライオリティ1のグループに発行制限があり、プライオリティ2およびプライオリティ3の各グループには発行制限がない。したがって、図9に戻り、プライオリティ1のグループについて、発行制限による遅延があるか否かを調べる（ステップS94）。なお、プライオリティ2およびプライオリティ3の各グループについては、発行制限がないため、発行制限による遅延の有無を調べる必要はない。

【0036】

発行制限による遅延の有無を調べるため、実VLIW命令数と最小VLIW命令数を求め、それらを比較する。図10に示すモデルでは、図13および図14

に示すように、命令3、命令5および命令6はI/Oスロットでしか発行することができないため、実VLIW命令数は3となる。それに対して、最小VLIW命令数は2であるので、前記(2)式を満たす。したがって、プライオリティ1のグループでは発行制限による遅延が発生することとなり、プライオリティ1のグループは発行制限の最適化処理をおこなう対象となる。本明細書では、発行制限の最適化処理をおこなう対象となるプライオリティのグループを最適化対象グループと呼ぶこととする。ここまで結果を図15にまとめて示す。

【0037】

つづいて、図9に戻り、ネック命令を取得する(ステップS95)。ここで、ネック命令とは、最適化対象グループ(図10に示すモデルではプライオリティ1のグループ)に含まれる命令の共通の先行命令のことであり、後述する重み付けをおこなう際の有効範囲を求めるために用いられる。最適化対象グループ内の命令は、ネック命令が発行された時点で互いに独立した関係、すなわち相互に依存しない関係となる。

【0038】

なお、ネック命令は最適化対象グループ内の全ての命令について共通である必要はなく、最適化対象グループ内的一部の命令群について共通の先行命令であってもよい。ある最適化対象グループに共通の先行命令が複数存在する場合には、それら複数の共通の先行命令のうち最小のプライオリティ値を有する命令をネック命令とし、そのネック命令のプライオリティ値をネックプライオリティ値と呼ぶ。図10に示すモデルでは、最適化対象グループはプライオリティ1のグループであり、また図16に示すように、ネック命令は命令1であり、ネックプライオリティ値は3である。

【0039】

つづいて、図9に戻り、前記(3)式に基づいて、逆プライオリティ計算をおこなう(ステップS96)。なお、図10に示すモデルのように、ネック命令が存在する場合には、ネック命令の逆プライオリティ値は1となる。つまり、図10に示すモデルではネック命令よりも先の先行命令は存在しないが、ネック命令よりも先に先行命令が存在する場合であっても、ネック命令よりも先の先行命令

については逆プライオリティ計算をおこなわない。

【0040】

図17に示すように、命令1はネック命令であるため、その逆プライオリティ値は1となる。命令2、命令4および命令6の逆プライオリティ値は2であり、命令3および命令5の逆プライオリティ値は3である。図17に示すDAGにおいて、各命令のノードを表す①～⑥の右下の数字は逆プライオリティ値を表す。

【0041】

つづいて、図9に戻り、最適化対象グループに含まれる複数の命令を逆プライオリティ値の小さい順に並べ替える（ステップS97）。その際、逆プライオリティ値が同じ場合には先行命令数の少ない順とし、逆プライオリティ値および先行命令数が同じ場合には行番号の早い順とし、さらに逆プライオリティ値、先行命令数および行番号が同じ場合には生成順番の早い順とする。したがって、図10に示すモデルでは、命令6、命令3、命令5の順となる（図17参照）。

【0042】

つづいて、各命令に対して重み付けをおこなう（ステップS98）。重み付けには第1の重み値アドバンテージと第2の重み値ウェイトの2種類がある。アドバンテージは、最適化対象グループからネック命令までの命令に付与される。ウェイトは、ネック命令よりも先の先行命令に付与される。なお、図10に示すモデルでは、ネック命令である命令1よりも先には命令がないため、ウェイトは発生しない。したがって、ここではアドバンテージについてのみ説明し、ウェイトについての説明は後述する。

【0043】

最適化対象グループ内に含まれる各命令に対し、実VLIW命令数から1を引いた値をアドバンテージの最初の値とし、ステップS97で並べ替えた順に一命令ごとにアドバンテージの値を1ずつ小さくする。最適化対象グループ内に含まれる各命令の先行命令のアドバンテージについては、その先行命令につづく後続命令のアドバンテージをそのまま継承する。後続命令が複数存在する場合には、最も大きいアドバンテージを継承する。

【0044】

図10に示すモデルでは、上述したように実V L I W命令数が3であり、かつステップS97で命令6、命令3、命令5の順で並べ替えをおこなったため、図18に示すように、命令6、命令3および命令5のアドバンテージは2、1および0（ゼロ）となる。そして、命令2のアドバンテージは命令3を継承するため1である。同様に、命令4のアドバンテージは命令5を継承するため0（ゼロ）である。命令1のアドバンテージは、命令2と命令4と命令6のうち最も大きいアドバンテージである命令6を継承するため2である。

【0045】

つづいて、図9に戻り、ステップS98でおこなった重み付けを考慮して、再度プライオリティ計算をおこなう（ステップS99）。ここでのプライオリティ計算は前記（4）式にしたがうが、後続命令のプライオリティ値に発行レイテンシとペナルティを加算したものは、ステップS91で求めた従来のプライオリティ値と同じである。つまり、新プライオリティ値は、今までのプライオリティ値に重み値を加算したものである。したがって、図19の上段に示すように、命令1の新プライオリティ値は5（=3+2）、命令2の新プライオリティ値は3（=2+1）、命令3の新プライオリティ値は2（=1+1）、命令4の新プライオリティ値は2（=2+0）、命令5の新プライオリティ値は1（=1+0）、命令6の新プライオリティ値は3（=1+2）である。

【0046】

図9に戻り、各命令の新プライオリティに基づいて、レディリストを生成し、スロット配置をおこない（ステップS100）、スケジューリングを終了する。各命令のスロット配置順は、新プライオリティ値が大きい順とし、新プライオリティ値が同じ場合には後続命令の数が少ない順とする。図10に示すモデルでは、命令1、命令6、命令2、命令3、命令4、命令5の順にスロット配置がおこなわれる。

【0047】

したがって、図19の下段右側に示すように、擬似機械テーブルにおいて、あるサイクルにおいてI0スロットから命令1が発行される。このときI1スロットからは命令が発行されない。そして、つぎのサイクルでI0スロットおよびI

1スロットからそれぞれ命令6および命令2が発行され、そのつぎのサイクルでI0スロットおよびI1スロットからそれぞれ命令3および命令4が発行され、さらにつぎのサイクルでI0スロットから命令5が発行される配置となる。つまり、命令1～命令6の6個の命令を4サイクルで発行することができる。

【0048】

ここで、比較のため、図19の下段左側に、重み付けをおこなう前の従来のプライオリティに基づいてスロット配置をおこなった結果を示す。この場合、命令1、命令2、命令4、命令3、命令5、命令6の順にスロット配置がおこなわれることとなり、全部で5サイクル必要となる。

【0049】

ところで、ステップS93で発行制限が発生しない場合、またはステップS94で発行制限による遅延がない場合には、ステップS91で計算した従来のプライオリティを用いてレディリストの生成およびスロット配置（ステップS100）をおこない、スケジューリングを終了する。

【0050】

つぎに、図20に示すモデルを用いてウェイトについて説明する。図20に示すモデルでは、命令1（図に①で示す）は命令2（図に②で示す）と依存関係にある。命令2は命令3（図に③で示す）と依存関係にある。命令3は命令4（図に④で示す）と依存関係にある。命令4は命令5（図に⑤で示す）と依存関係にある。命令6（図に⑥で示す）は命令7（図に⑦で示す）と依存関係にある。命令7は命令8（図に⑧で示す）と依存関係にある。命令8は命令4、命令9（図に⑨で示す）および命令11（図に○11示す）と依存関係にある。命令9は命令10（図に○10示す）と依存関係にある。

【0051】

図20に示すモデルでは、最適化対象グループに含まれる命令は、命令5と命令10と命令11であり、ネック命令は命令8である。命令3および命令8の逆プライオリティ値は1である。命令4、命令9および命令11の逆プライオリティは2である。命令5および命令10の逆プライオリティ値は3である。

【0052】

ウェイトは、ネック命令よりも先の先行命令、すなわちネックプライオリティよりも大きなプライオリティ値を有する命令に対して付与される。ウェイトは、そのウェイトが付与される命令のすぐ後の最適化対象グループの実VLIW命令数に等しい。そして、先行命令のウェイトは、後続命令のウェイトをそのまま継承する。したがって、図20に示すモデルでは、最適化対象グループ（命令5、命令10および命令11）の実VLIW命令数は3であるため、図21に示すように、ネック命令（命令8）よりも先にある命令2および命令7のウェイトは3となる。命令2および命令7よりも先にある命令1および命令6については、命令2および命令7のウェイトを継承するため、ウェイトは3となる。

【0053】

ここで、ウェイトは、別の最適化対象グループに起因して新たなウェイトが付与されると、それを加算していく。これについて、たとえば図22に示すモデルを用いて具体的に説明する。図22に示すモデルでは、命令1（図に①で示す）は命令2（図に②で示す）と依存関係にある。命令2は命令3（図に③で示す）と依存関係にある。命令3は命令4（図に④で示す）と依存関係にある。命令4は命令5（図に⑤で示す）と依存関係にある。命令5は命令6（図に⑥で示す）と依存関係にある。命令6は命令7（図に⑦で示す）と依存関係にある。

【0054】

命令7は命令8（図に⑧で示す）と依存関係にある。命令9（図に⑨で示す）は命令10（図に○10で示す）と依存関係にある。命令10は命令3および命令11（図に○11で示す）と依存関係にある。命令11は命令12（図に○12で示す）と依存関係にある。命令12は命令13（図に○13で示す）と依存関係にある。命令13は命令14（図に○14で示す）と依存関係にある。命令14は命令7および命令15（図に○15で示す）と依存関係にある。命令15は命令16（図に○16で示す）と依存関係にある。

【0055】

このモデルでは、第1の最適化対象グループに含まれる命令は、命令8と命令16であり、この第1の最適化対象グループに対するネック命令は命令14である。また、このモデルでは、命令4と命令12は、命令10をネック命令とする

第2の最適化対象グループに含まれる。

【0056】

図22に示すモデルでは、命令8および命令16の従来のプライオリティ値は1である。そして、従来のプライオリティ値は、命令8から命令1に向かって、また命令16から命令9に向かって順に1ずつ大きくなり、命令1および命令9のプライオリティ値は8である。このようなモデルに対して、逆プライオリティ値は、図23に示すように、命令8と命令16と命令4と命令12では3であり、命令7と命令15と命令3と命令11では2であり、命令6と命令14と命令2と命令10では1である。

【0057】

図22に示すモデルに対して付与されるウェイトは、図24に示すように、命令6～命令8および命令14～命令16では0（ゼロ）である。第1の最適化対象グループ内の命令8および命令16の実VLIW命令数は2であるため、ネック命令である命令14よりも先の先行命令である命令5および命令13のウェイトは2となる。このウェイトの値はさらに先の先行命令に継承されるため、命令2～命令4および命令10～命令12のウェイトも2となる。さらには、第1の最適化対象グループに起因して付与された2のウェイトはさらに先の命令1および命令9にも継承される。

【0058】

また、第2の最適化対象グループ内の命令4および命令12の実VLIW命令数は2であるため、この第2の最適化対象グループに対応するネック命令である命令10の先行命令である命令1および命令9には、新たに第2の最適化対象グループに起因する2のウェイトが付与される。したがって、命令1および命令9のウェイトは、第1の最適化対象グループに起因して付与された2と、第2の最適化対象グループに起因して付与された2とを足して4となる。なお、図24に示すDAGにおいて、各命令のノードを表す①～○16の右下の式のうち、「+」を挟んで左側の数字はウェイトを表し、右側の数字はアドバンテージを表す。各命令に対して付与される実際の重み値は、各命令のウェイトとアドバンテージを加算したものである。

【0059】

つぎに、本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明する。つぎの（1）～（3）のいずれかの条件に該当する場合には、発行制限の最適化性能が低下するおそれがあるため、発行制限の最適化処理を実施しない。（1）最適化対象グループの命令に先行命令が存在しない。（2）ネック命令が存在するが、最適化対象グループのプライオリティとネックプライオリティとの間に他の命令が全く存在しない。（3）ネック命令が存在するが、最適化対象グループのプライオリティとネックプライオリティとの間に、最適化対象グループの先行命令ではない命令が存在する。

【0060】

たとえば図25に示すモデルでは、命令1（図に①で示す）は命令2（図に②で示す）と依存関係にある。命令3（図に③で示す）は命令4（図に④で示す）と依存関係にある。命令5（図に⑤で示す）は命令6（図に⑥で示す）と依存関係にある。命令2、命令4および命令6のプライオリティ値は1である。命令1、命令3および命令5のプライオリティ値は2である。そして、最適化対象グループに含まれる命令は、命令1と命令3と命令5である。このモデルは上述した（1）の条件に該当するため、発行制限の最適化処理をおこなわない。

【0061】

また、図26に示すモデルでは、命令1（図に①で示す）は命令2（図に②で示す）と依存関係にある。命令3（図に③で示す）は命令4（図に④で示す）および命令5（図に⑤で示す）と依存関係にある。命令2、命令4および命令5のプライオリティ値は1である。命令1および命令3のプライオリティ値は2である。そして、最適化対象グループに含まれる命令は、命令2と命令4と命令5であり、これに対するネック命令は命令3である。このモデルは上述した（2）の条件に該当するため、発行制限の最適化処理をおこなわない。

【0062】

また、図27に示すモデルでは、命令1（図に①で示す）は命令2（図に②で示す）と依存関係にある。命令2は命令3（図に③で示す）と依存関係にある。

命令3は命令4（図に④で示す）と依存関係にある。命令5（図に⑤で示す）は命令6（図に⑥で示す）、命令9（図に⑨で示す）および命令12（図に○12で示す）と依存関係にある。命令6は命令7（図に⑦で示す）と依存関係にある。命令7は命令8（図に⑧で示す）と依存関係にある。

【0063】

命令9は命令10（図に○10示す）と依存関係にある。命令10は命令11（図に○11示す）と依存関係にある。命令12は命令13（図に○13示す）と依存関係にある。命令4、命令8、命令11および命令13のプライオリティ値は1である。命令3、命令7および命令10のプライオリティ値は2である。命令2、命令6、命令9および命令12のプライオリティ値は3である。命令1および命令5のプライオリティ値は4である。そして、最適化対象グループに含まれる命令は、命令3と命令7と命令10であり、これに対するネック命令は命令5である。このモデルは上述した（3）の条件に該当し、最適化対象グループのプライオリティとネックプライオリティとの間に、最適化対象グループ内の命令と依存関係がない命令、図示例では命令12が存在するため、発行制限の最適化処理をおこなわない。

【0064】

上述した実施の形態によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群について発行制限による遅延があるか否かを調べるため、全命令のノードに対して並列動作抑制関係を調べるのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【0065】

また、上述した実施の形態によれば、逆プライオリティ計算およびアドバンテージを付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、全命令のノードに対して並列動作抑制数を計算するのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【0066】

さらに、上述した実施の形態によれば、最適化対象グループからネック命令ま

での命令にアドバンテージを付与し、ネック命令よりも先の先行命令にウェイトを付与するため、命令に付与する重み値が並列動作抑制数だけである場合に比べて、目的コードの並列度を向上させることができる。

【0067】

(付記1) 命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【0068】

(付記2) 命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

同じプライオリティ値の命令間に発行制限による遅延があるか否かを調べる工程と、

発行制限による遅延がある場合に、前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

前記新プライオリティ値に基づいて命令の発行順序を決定し、前記各命令のスロット配置をおこなう工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【0069】

(付記3) 発行制限による遅延がある命令群を最適化対象グループとし、当該最適化対象グループに含まれる複数の命令に共通の先行命令をネック命令とし、当該ネック命令と前記最適化対象グループとの間で前記逆プライオリティ値

を求めることを特徴とする付記2に記載のコンパイラ並列化スケジュール方法。

【0070】

(付記4) 前記重み付けは、前記最適化対象グループから前記ネック命令までの命令に付与される第1の重み値と、前記ネック命令よりも先の先行命令に付与される第2の重み値とからなることを特徴とする付記3に記載のコンパイラ並列化スケジュール方法。

【0071】

(付記5) 前記最適化対象グループに含まれる複数の命令に対して、前記逆プライオリティ値の小さい順、前記逆プライオリティ値が同じ場合には前記先行命令数の少ない順、前記逆プライオリティ値と前記先行命令数が同じ場合には行番号の早い順、または前記逆プライオリティ値と前記先行命令数と前記行番号が同じ場合には生成順番の早い順に、優先順位を設定し、その優先順位にしたがって、前記第1の重み値を決定することを特徴とする付記4に記載のコンパイラ並列化スケジュール方法。

【0072】

(付記6) 前記優先順位にしたがって、第1番目の命令の第1の重み値を、前記最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数から1を引いた値とし、第2番目以降の命令の第1の重み値を前記命令数から1を引いた値から順次1ずつ小さくした値とすることを特徴とする付記5に記載のコンパイラ並列化スケジュール方法。

【0073】

(付記7) 前記最適化対象グループ内の各命令に対する先行命令の第1の重み値を、当該先行命令につづく後続命令の第1の重み値をそのまま継承した値とし、前記後続命令が複数存在する場合には最も大きい第1の重み値を継承した値とすることを特徴とする付記5または6に記載のコンパイラ並列化スケジュール方法。

【0074】

(付記8) 前記ネック命令の先行命令の第2の重み値を、前記ネック命令に対応する最適化対象グループ内の命令を実際の発行制限を考慮して発行するの

に要する命令数に等しい値とすることを特徴とする付記4に記載のコンパイラ並列化スケジュール方法。

【0075】

(付記9) 前記ネック命令の先行命令の第2の重み値を、前記ネック命令に対応する最適化対象グループとは別の最適化対象グループに起因して新たな第2の重み値が発生した場合には、当該第2の重み値を加算した値とすることを特徴とする付記4に記載のコンパイラ並列化スケジュール方法。

【0076】

(付記10) 同じプライオリティ値の命令間に発行制限がある場合、同じプライオリティ値の命令を実際の発行制限にしたがって発行するのに要する命令数と、同じプライオリティ値の命令を発行制限がないと仮定して発行するのに要する命令数とを求め、前記両命令数を比較して、実際の発行制限にしたがって発行するのに要する命令数の方が大きい場合に発行制限による遅延があると判断することを特徴とする付記2～9のいずれか一つに記載のコンパイラ並列化スケジュール方法。

【0077】

(付記11) 前記最適化対象グループの命令に先行命令が存在しない場合、最適化対象グループのプライオリティ値とネック命令のプライオリティ値との間に他の命令が全く存在しない場合、または最適化対象グループのプライオリティ値とネック命令のプライオリティ値との間に、最適化対象グループの先行命令ではない命令が存在する場合のいずれか一つ以上に該当する場合、前記逆プライオリティ値の計算工程、前記重み付け工程および前記新プライオリティ値の計算工程をおこなわずに、最初に求めたプライオリティ値に基づいて命令の発行順序を決定し、各命令のスロット配置をおこなうことを特徴とする付記3に記載のコンパイラ並列化スケジュール方法。

【0078】

【発明の効果】

本発明によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群につい

て発行制限による遅延があるか否かを調べるとともに、逆プライオリティ計算および第1の重み値を付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、より高速に目的コードの並列化処理をおこなうことができる。さらに、本発明によれば、最適化対象グループからネック命令までの命令に第1の重み値を付与し、ネック命令よりも先の先行命令に第2の重み値を付与するため、目的コードの並列度を向上させることができる。

【図面の簡単な説明】

【図1】

本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャートである。

【図2】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図3】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図4】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図5】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図6】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図7】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図8】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図9】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に示すフローチャートである。

【図10】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図11】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図12】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図13】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図14】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図15】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図16】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図17】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図18】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図19】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図20】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図21】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図22】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図23】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図24】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図25】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図である。

【図26】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図である。

【図27】

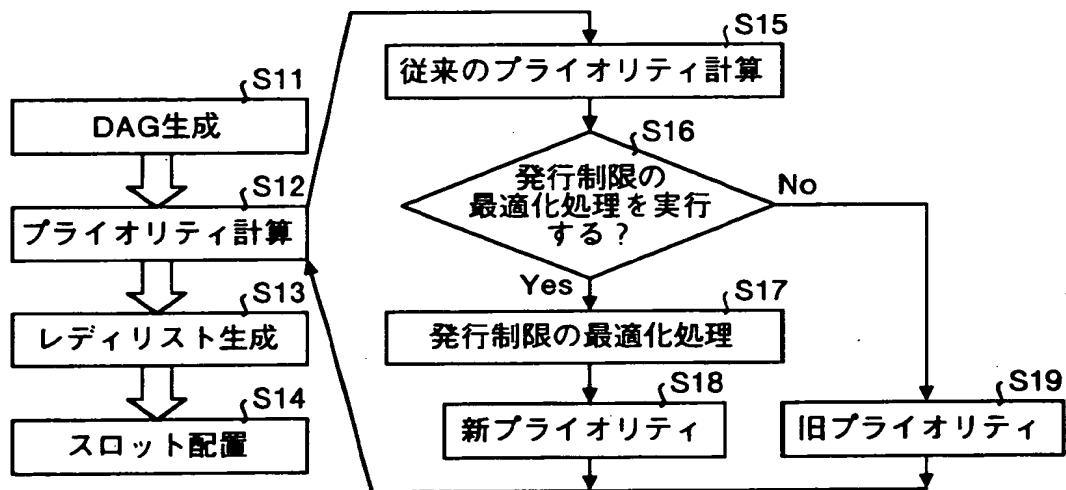
本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化

処理をおこなわない場合について説明するための模式図である。

【書類名】 図面

【図1】

本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャート



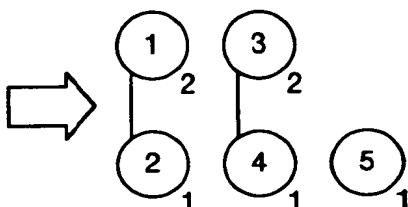
【図2】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図

サンプルソース

- ① move gr6,cnt('A')
- ② store @(gr4,# 1),gr6
- ③ move gr7,cnt('B')
- ④ store @(gr4,# 2),gr7
- ⑤ store @(gr4,# 3),gr0

DAGとプライオリティ



【図3】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための図表

プライオリティ	命令
2	①③
1	②④⑤

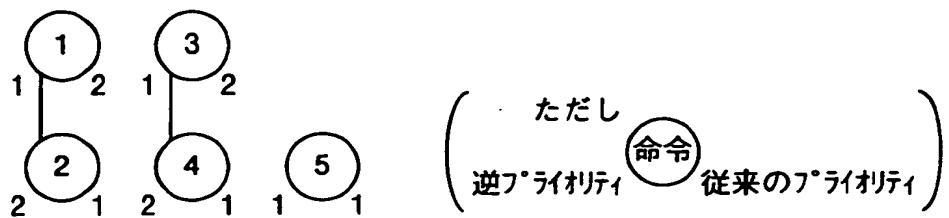
【図4】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための図表

プライオリティ	実VLIW 命令数	最小VLIW 命令数	発行制限対応 処理必要有無
2	1	1	なし
1	3	2	あり

【図5】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための模式図



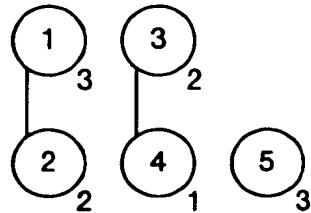
【図6】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための図表

命令	逆プライオリティ値	生成順番	重み
②	2	1	1
④	2	2	0
⑤	1	3	2

【図7】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための模式図



【図8】

本発明にかかるコンパイラ並列化スケジュール方法の概要を
説明するための模式図

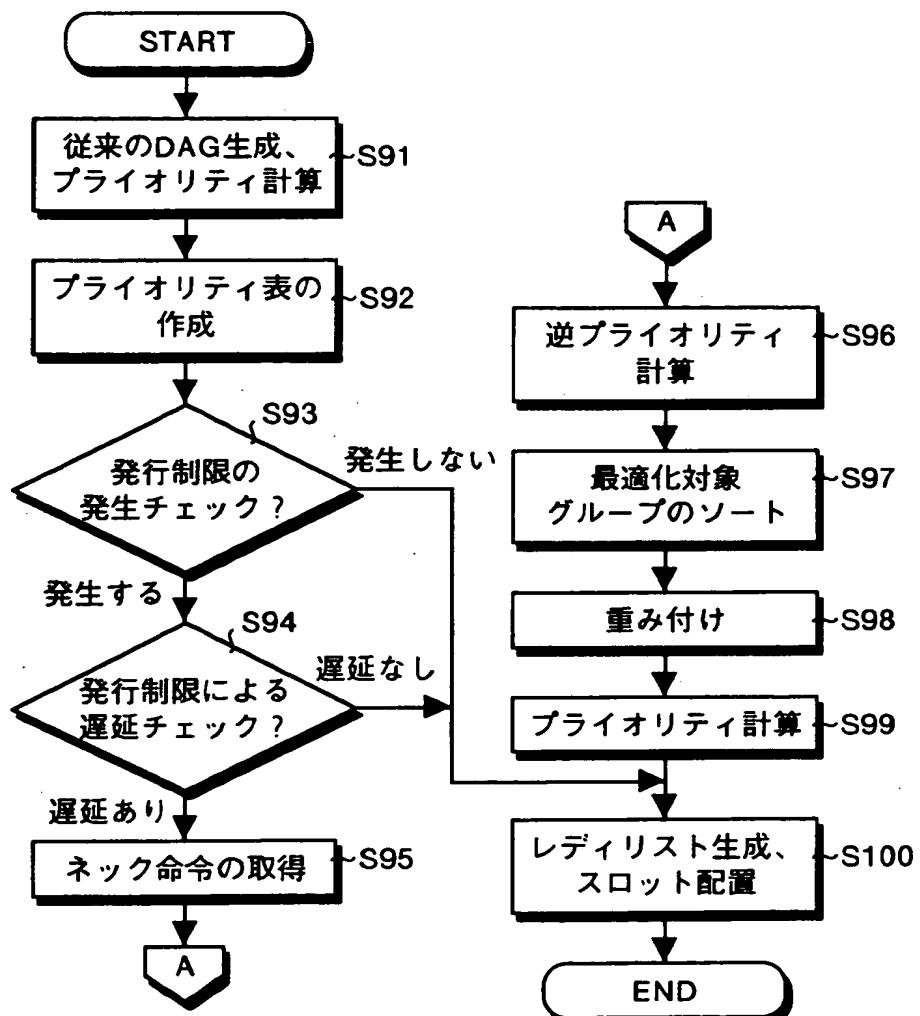
⑤→①→②→③→④の
順にスロット配置が
行われる

擬似機械テーブル

	10スロット	11スロット
α	5	1
$\alpha + 1$	2	3
$\alpha + 2$	4	

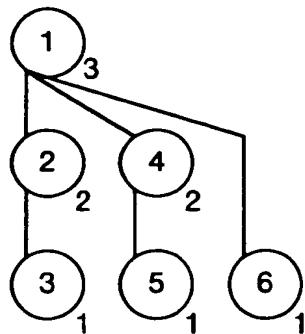
【図9】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に示す
フローチャート



【図10】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図



【図11】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための図表

プライオリティ	命令リスト
1	③,⑤,⑥
2	②,④
3	①

【図12】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための図表

プライオリティ	命令リスト	発行制限チェック結果
1	③,⑤,⑥	あり
2	②,④	なし
3	①	なし

【図13】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図

最小VLIW命令数：2
(発行制限なしの場合のVLIW命令数)

	I0スロット	I1スロット
VLIW1	3	5
VLIW2	6	



実VLIW命令数：3
(発行制限ありの場合のVLIW命令数)

	I0スロット	I1スロット
VLIW1	3	
VLIW2	5	
VLIW3	6	

【図14】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための図表

発行制限なしの場合の VLIW命令数 (最小VLIW命令数)	実際に発行に必要な VLIW命令数 (実VLIW命令数)
2個	3個

【図15】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための図表

プライオリティ	命令リスト	発行制限チェック結果	遅延チェック
1	③,⑤,⑥	あり	あり
2	②,④	なし	—
3	①	なし	—

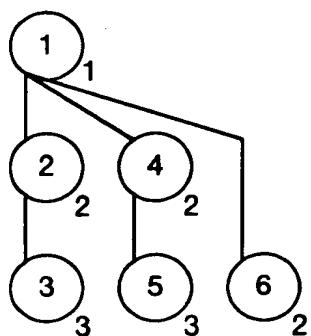
【図16】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための図表

ネック命令	ネックプライオリティ
①	3

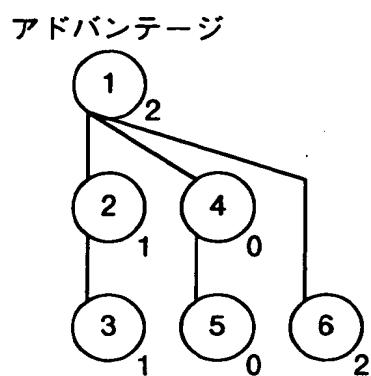
【図17】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図

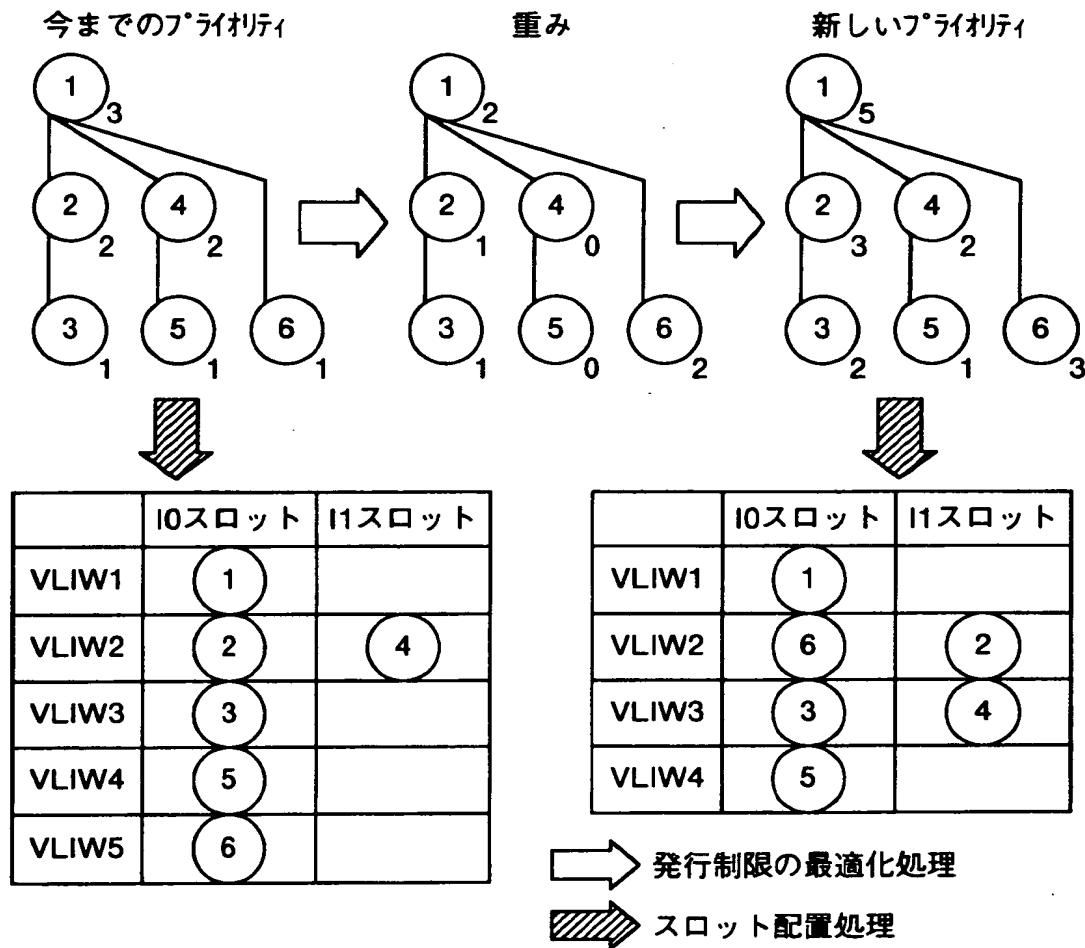


【図18】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図



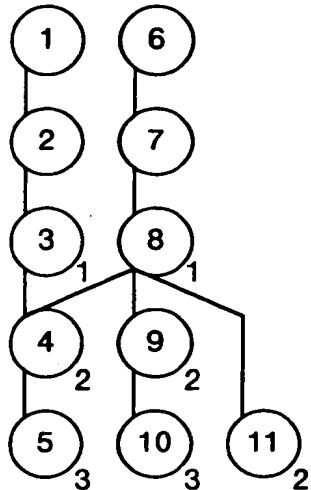
【図19】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図

【図20】

本発明にかかるコンバイラ並列化スケジュール方法を詳細に
説明するための模式図

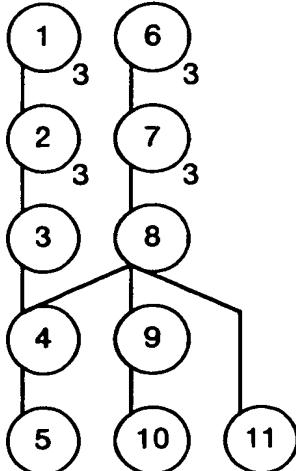
逆プライオリティ



【図21】

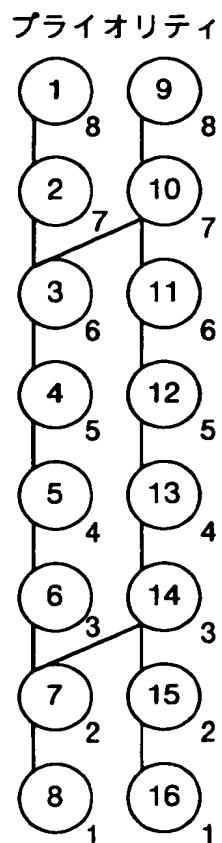
本発明にかかるコンバイラ並列化スケジュール方法を詳細に
説明するための模式図

ウェイト



【図22】

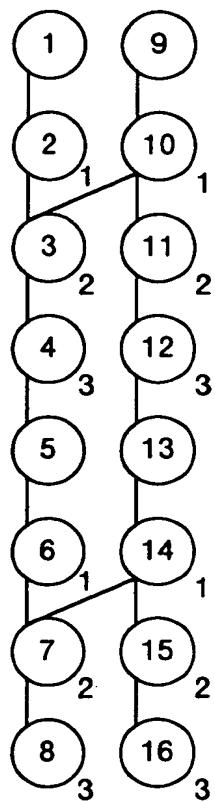
本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図



【図23】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図

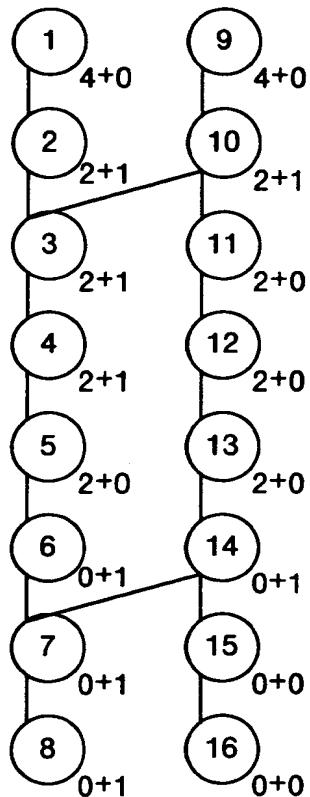
逆プライオリティ



【図24】

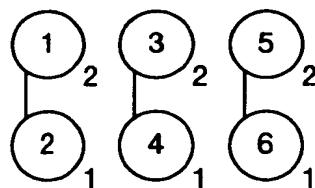
本発明にかかるコンパイラ並列化スケジュール方法を詳細に
説明するための模式図

重み(Weight+advantage)



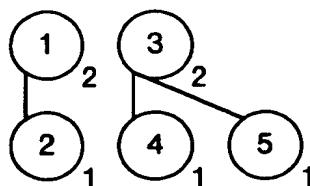
【図25】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図



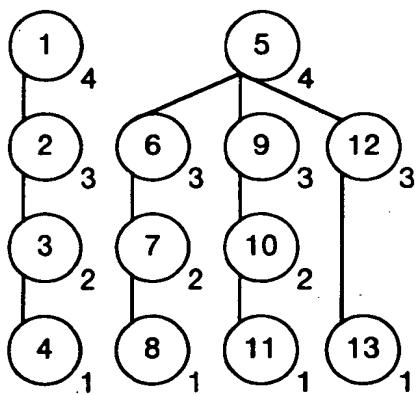
【図26】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図



【図27】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の
最適化処理をおこなわない場合について説明するための模式図



【書類名】 要約書

【要約】

【課題】 並列に動作し得る複数の演算ユニットを持つ計算機用の目的コードを生成するコンパイラ並列化スケジュール方法において、目的コードの並列度を向上させるとともに、より高速に目的コードの並列化処理をおこなうこと。

【解決手段】 従来のプライオリティ値ごとに命令を分類し、プライオリティ値が同じ命令間で発行制限があるかを調べ、発行制限がある命令群について発行制限による遅延があるかを調べ、発行制限による遅延がある命令群よりなる最適化対象グループから、その最適化対象グループ内の命令に共通の先行命令であるネック命令までの命令に対して逆プライオリティ計算をおこない、その逆プライオリティ値に基づき、再度プライオリティ計算をおこない、スロット配置をおこなう。

【選択図】 図9

出願人履歴情報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社